Network Explorer: Design, Implementation, and Real World Deployment of a Large Network Visualization Tool

John Alexis Guerra-Gomez, Aaron Wilson*, Juan Liu†, Dan Davies‡, Peter Jarvis§and Eric Bier¶
Palo Alto Research Center
john.guerra@gmail.com

ABSTRACT

This paper describes the process of design, implementation, and real world deployment of a web-based network exploration tool called Network Explorer. We designed Network Explorer based on the expressed needs of our clients and later deployed it as part of a larger system for fraud detection in health care. Our implementation of Network Explorer provides visual interactive access to large-scale network data. As part of the Network Explorer tool we contribute a dynamic group-in-a-box implementation for laying out clusters, and a node navigator widget that aids in the exploration of large networks. We are also contributing two open source components of the Network Explorer for the community to reuse: an in-browser clustering library, and the dynamic group-in-a-box algorithm. We have evaluated the network explorer tool in multiple real-world environments including the fraud detection setting above.

CCS Concepts

•**Human-centered computing** → **Graph drawings;** *Visual analytics; Empirical studies in visualization;*

Keywords

Network Visualization of Large Networks; Industry Visualization Deployment

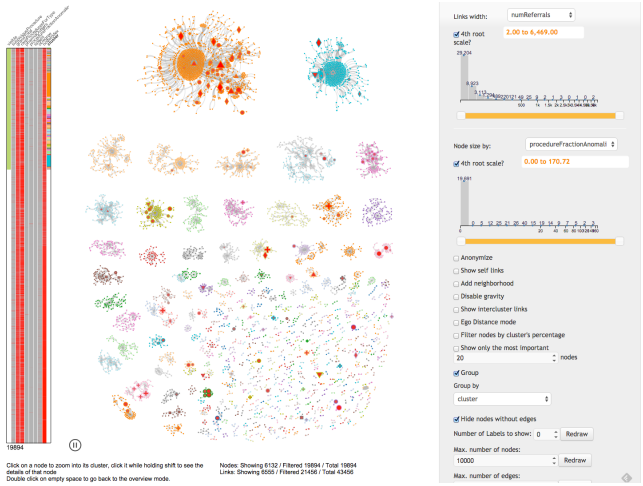

Figure 1: Network of 19,894 doctors and pharmacies that interact through prescriptions in a medical system. Network Explorer's rank-by-feature shows the top 6,132 most important nodes for the analyst, according to the level of suspiciousness of each node and its connectivity. Network Explorer offers the option for analysts to hide edges between clusters for legibility.

1. INTRODUCTION

Many research contributions from the information visualization field could greatly improve data analysis processes in the industry and the government. However, making real world implementations and deployments of such systems is a daunting task. Many hurdles need to be jumped before a research prototype can be accessed by the users that so much could benefit from it. On the other hand, real world deployments of state of the art information visualization techniques as products are a remarkable way of validating the applicability and utility of such techniques beyond what can be achieved during constrained lab experiments.

In this paper we describe the design, implementation and real world deployment of Network Explorer, a web-based network exploration tool, that implements state of the art information visualization techniques. Moreover, we have demonstrated the generality of Network Explorer by applying it to multiple application domains including fraud detection in health care, police chat networks, and drug sale networks, however for lack of space we cannot present more details on this paper. We contribute our finding during deployment as evidence of what worked and what didn't for us in our products, with the goal of informing future research in the area of network visualization.

We deployed Network Explorer as an integrated module of

*aaron.cr.wilson@gmail.com
†JuanLiuWU01@yahoo.com
‡Dan.Davies@parc.com
§peter.jarvis@gmail.com
¶bier@parc.com

AVI '16, June 07 - 10, 2016, Bari, Italy

ISBN 978-1-4503-4131-8/16/06…$15.00

DOI: http://dx.doi.org/10.1145/2909132.2909281

a fraud, waste, and abuse (FWA) detection tool, an application that helps analysts identify crimes in health insurance claims. Network Explorer was deployed, and has been used since then, to find suspicious clusters and providers out of the millions of health insurance claims reported in health care systems. Network Explorer is available in the multiple sites where the FWA tool is used. Moreover we evaluated Network Explorer in two other problem domains, analyzing police officer's chat networks and drug sales networks. Our in-situ think aloud evaluation of user's usage showed that our users were able to find suspicious networks and connections in ways that weren't possible before. The rest of this paper presents the related work used to inspire our system, the modes of use that we found worked best for our users, and the technical contributions developed to power those modes.

2. RELATED WORK

The abundance of problems involving network analysis has inspired much work in the information visualization community. The surveys conducted by Herman et al. [17], Heer et al. [12] and VonLandesberger et al. [30] discuss the landscape of the alternatives for visualizing networks. This work has triggered the development of many general-purpose applications for visualizing networks such as Gephi, Cytoscape [26], Vizter [13], Visone [19], Lanet-vi [4], Pajek [3], NodeTrix [16], NodeXL [5], Tulip [2] and GCV [28]. Researchers have also worked on the problem of clustered network visualizations [1]. Also, many programming libraries support the creation of network visualization such as Prefuse [15], Flare, Protovis [6], D3.js [7], Sigma.js, NetworkX [11], Jung. Even companies such as Linkurious specialize in visualizing networks. We designed, implemented and deployed Network Explorer to address the specific needs of our customers, implementing selected best practices found in the state of the art, and extending their features when needed. A new network visualization tool was necessary because our users required a solution that was deeply integrated with their current tools, adapted for their data needs and that didn't require the specialized training.

Large network visualization is a hard and relevant problem that has been tackled in the past by many authors [3, 9, 10, 22]. To allow the exploration of such networks, we built upon the concept of rank-by-feature [23], to build the rank-by-relevance algorithm that selects the most relevant nodes of the network to be visualized.

Shneiderman's visualization mantra of "Overview first, zoom and filter, then details-on-demand" [27] dictates that a visualization systems should provide a good overview of the data, support zoom and filtering options, and offer details when the user requests them. Following the infoviz mantra, the Network Explorer features an overview mode that allows users to obtain an idea of the network's structure. It also provides controls to filter nodes and zoom into clusters, and offers detailed information at users' request.

Also, for the overview mode, we built a network clustering library[1] on top of the Clauset, Newman and Moore algorithm [8] implemented by Robin W. Spencer in his website Scaled Innovation[2]. To visualize the clusters inferred by the algorithm, we developed a dynamic group-in-a-box implementation (released as an open source library[3]). Our implementation improves upon the original [25] by allowing a tampered interactions between clusters, which distribute nodes connecting clusters closer to their connections. Being interactive, our group-in-a-box implementation also allows users to track the transformation by means of animation.

The ego-centric mode was inspired by previous work on the degree of interest graphs [29], Heer et al's degree of interest tree [14], and Plaisant et al's Space Tree [24]. Our ego-centric mode builds upon their research by adding an ego-distance layout, combined with a constrained force-directed layout, and the ability to pin nodes in specific areas of the screen.

Finally, previous work has suggested that curved edges can be problematic for network visualizations [18, 20]. Despite that, in our deployments our users shown strong preference for curved edges, because of their ability to show independent directional connections (A->B and B->A with different edge values), and because they found them more visually appealing. Further work may be required to explore in which specific tasks and environments curved edges could be detrimental for users.

3. THE NETWORK EXPLORER

Network Explorer was designed to help our customers, fraud detection analysts, and officers. Specifically our users wanted: 1. To find important clusters of nodes (Which correspond **overview** and **topology** tasks from the task taxonomy of graph visualization [21]) and 2. Identify the important actors, nodes, within the network (**Attribute-based** and **browsing** tasks from the taxonomy).

To address these tasks, Network Explorer provides two main modes of use: the overview mode and the ego-centric mode. The overview mode enables users to get an eagle-eye picture of the network providing a view of the macro properties of the network. Users can achieve this by means of the rank-by-relevance framework that selects the most relevant nodes to visualize, clustering algorithms that identify communities in the network, and a dynamic group-in-a-box implementation that highlights those communities. The second mode of use, the ego-centric mode, allows users to select one or more target nodes and interactively explore their connections in order to visualize important details of their relationships.

The rest of this section describes each mode in more detail along with the features that power them.

3.1 Overview mode

When analyzing a network, users commonly want to understand it as a whole. For small enough networks, force-directed layouts can address this need. However, with just a few hundred of nodes, the results are commonly a meaningless "hairball". To address this, the Network Explorer includes an overview mode that helps users get an idea of the structure of the network by: 1) computing the inherent communities 2) visualizing them using dynamic group-in-a-box algorithm. 3) Selecting the most significant nodes to visualize using a rank-by-relevance framework. 4) Allowing navigation into communities and computation of sub-communities on demand. 5) Allowing filtering by node and

[1]https://github.com/john-guerra/netClusteringJs

[2]http://scaledinnovation.com/analytics/communities/communities.html

[3]https://github.com/john-guerra/forceInABox

edge properties. and Finally 6) Allowing the navigation of nodes in a tabular display.

The rest of this section describes most of these features in detail.

Clustering. Making sense of a picture of undifferentiated nodes and arcs can be difficult. The Network Explorer takes advantage of two clustering algorithms, the server-sided (proprietary) scalable Goal Directed Clustering algorithms (GDC), and NetClustering.js a JavaScript library that runs directly on the browser but is limited in the number of nodes it can process. The GDC is capable of processing millions of nodes, allowing users to explore large networks. However, since it runs on the server side it needs to be run at least once as a pre-processing step per dataset. On the other hand, NetClustering.js can be used on smaller networks directly on the browser, allowing users to recompute clusters on specific subsets of the network (fast enough to recompute thousands of nodes in a few seconds). NetClustering.js was built on top of open source software and has been made available for the scientific community to reuse.

Group in a Box. Understanding the overall structure of a network can be difficult, even with a network visualization color-coded by cluster. The Group-in-a-box layout algorithm introduced in [25] provides better structure understanding by distributing each community into its own box. Then, each group is laid out independently within its box. The original implementation of group-in-a-box was made for NodeXL [5] and therefore was non-interactive. Users could see the original network and then the final result, but they would miss the process that build it. The dynamic group-in-a-box algorithm presented in this paper builds upon the original by providing smooth and interactive animations that intuitively transform a plain force-directed network into a group-in-a-box version. Moreover, beyond the original implementation, the dynamic group-in-a-box algorithm can be configured to keep some interactions between clusters, allowing nodes connecting multiple clusters to be repositioned closer to the connected clusters.

Rank by relevance. Visualizing networks of hundreds of thousands of nodes is a daunting task. On the one hand there is a limit on how much the human brain can process at once, and on the other hand the browser starts getting too slow when drawing and animating more than a few thousand nodes at a time. Because of these, creating interactive visualizations of hundreds of thousands of nodes proves impractical. Despite that, the real world problems faced by the Network Explorer, such as the network of prescriber and pharmacies, go easily over the two hundred thousand nodes ceiling. To address problems of this size, we introduce the rank-by-relevance framework. Using this framework a subset of the most relevant nodes of the network is selected and visualized for the user to grasp how the network looks.

The rank-by-relevance framework works by filtering nodes by one or more of the following methods: 1) Rank nodes/edges by one or many of their parameters, e.g. Show only the most suspicious doctors. 2) Compute the inherent communities using the Goal Directed Clustering, then show a sample of the most representative nodes (according to their parameters) on each cluster, e.g. Show the 10% of the nodes on each cluster. 3) Show only the nodes in one cluster. 4) Show only nodes with connections. 5) Display the neighborhood of selected significant nodes. i.e. Show the nodes that wouldn't classify to be in the top list of selected nodes, but that are connected to one that is. 6) Force the display of nodes explicitly requested by the user, using the user interface.

Node navigator. The Node Navigator is a visualization widget that helps users understand what part of the network has been selected by the rank-by-relevance framework. The Node Navigator, shown in figure 1 left side, uses a vertical rectangle to represent the total number of nodes in the network. The height of the rectangle is divided among the total number of nodes, which are represented as rows or horizontal bars. Of each node, at least three attributes or columns are represented with colors: visibility, ranking-attribute and cluster, however the Node Navigator can show more numeric attributes if necessary. Green is used for visible nodes, while white represents hidden ones. For the ranking attribute (i.e. the node attribute used to rank and select which nodes to show) and other numeric attributes, independent scales ranging from gray to red are used. Finally, a scale of categorical colors is used for the clusters. Users can click on each one of the attributes to sort the Node Navigator by that attribute, which can be very useful to understand the distribution of nodes by cluster, how many nodes are being displayed, or the distribution of values of the ranking-attribute.

3.2 Ego-centric mode

Exploring the specific connections of a node (or group of nodes) in the network is the second most requested requirement from our users. As an example, one of our users was interested in understanding which pharmacies were driving the business from a specific prescriber, and identifying which other prescribers were heavy customers of said pharmacy. To meet this requirement, the Network Explorer includes an ego-centric mode that let's users select one or more target nodes, and interactively explore their connections. All of the features available in the overview mode also work in the ego-centric mode, including filtering, dynamic group-in-a-box and rank-by-relevance. On top of that, the ego-centric mode also features an extra ego-distance layout, which allows users to visualize the nodes that are one step away from the target nodes, the ones that are two steps away, etc.

The ego-distance layout allow users to navigate nodes at two, three or n hops from a starting node. This layout distributes nodes horizontally on the screen according to their shortest-path distance from the target nodes. The target nodes are placed first on the left side of the screen, and then distributed at n steps to the right. A constrained force system fixes the x axis, distributes the nodes across the y axis, avoids overlapping, and keeps the most tightly connected nodes in the center. Initially only the nodes at distance one are displayed, but users can select any node of interest and expand/contract their connections interactively. To reduce screen clutter, a degree-of-interest (DOI) measure, based in the rank-by-relevance algorithm, is used to rank the nodes. Only the top n nodes according to the DOI are displayed initially, with the option to add more at user's request. The DOI measure takes into account each node's attributes as well as the attributes of its edges. However, the specific formula of the DOI measure depends on the application domain.

4. ACKNOWLEDGMENTS

We the users of our tools for all of their thoughtful feedback; Robin W. Spencer for this implementation of the Clauset

et al. clustering algorithm; and Mike Bostock for D3.

5. REFERENCES

[1] J. Abello, F. van Ham, and N. Krishnan. ASK-GraphView: A large scale graph visualization system. *IEEE transactions on visualization and computer graphics*, 12(5):669–76, jan 2006.

[2] D. Auber, D. Archambault, R. Bourqui, A. Lambert, M. Mathiaut, P. Mary, M. Delest, J. Dubois, and G. Melançon. The Tulip 3 Framework: A Scalable Software Library for Information Visualization Applications Based on Relational Data. Technical report, jan 2012.

[3] V. Batagelj and A. Mrvar. *Pajek - analysis and visualization of large networks*. Springer, 2004.

[4] M. G. Beiró, J. I. Alvarez-Hamelin, and J. R. Busch. A low complexity visualization tool that helps to perform complex systems analysis. *New Journal of Physics*, 10, 2008.

[5] E. M. Bonsignore, C. Dunne, D. Rotman, M. Smith, T. Capone, D. L. Hansen, and B. Shneiderman. First Steps to NetViz Nirvana: Evaluating Social Network Analysis with NodeXL. 2009.

[6] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *Visualization and Computer Graphics, ...*, 2009.

[7] M. Bostock, V. Ogievetsky, and J. Heer. D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, dec 2011.

[8] A. Clauset, M. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 2004.

[9] N. Elmqvis, T. N. Do, H. Goodell, N. Henry, and J. D. Fekete. ZAME: Interactive large-scale graph visualization. In *IEEE Pacific Visualisation Symposium 2008, PacificVis - Proceedings*, pages 215–222, 2008.

[10] S. Hachul and M. Jünger. An experimental comparison of fast algorithms for drawing general large graphs. *Graph Drawing*, 2006.

[11] A. Hagberg, D. Schult, and P. Swart. Networkx. High productivity software for complex networks. *... strá nka https://networkx. ...*, 2013.

[12] J. Heer, M. Bostock, and V. Ogievetsky. A tour through the visualization zoo. *Communications of the ACM*, 53(6):59–67, 2010.

[13] J. Heer and D. Boyd. Vizster: Visualizing online social networks. *... Visualization, 2005. INFOVIS 2005. IEEE ...*, 2005.

[14] J. Heer and S. K. Card. DOITrees revisited: scalable, space-constrained visualization of hierarchical data. In *Proceedings of the working conference on Advanced visual interfaces - AVI '04*, page 421, New York, New York, USA, may 2004. ACM Press.

[15] J. Heer, S. K. Card, and J. A. Landay. prefuse: a toolkit for interactive information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05*, page 421, 2005.

[16] N. Henry, J. Fekete, and M. McGuffin. NodeTrix: a hybrid visualization of social networks. *Visualization and Computer ...*, 2007.

[17] I. Herman, G. Melançon, M. S. Marshall, G. Melancon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1):24–43, 2000.

[18] D. Holten, P. Isenberg, J.-D. Fekete, and J. V. Wijk. Performance Evaluation of Tapered, Curved, and Animated Directed-Edge Representations in Node-Link Graphs. Technical report, sep 2010.

[19] M. Jünger and P. Mutzel, editors. *Graph Drawing Software*. Mathematics and Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[20] Kai Xu, C. Rooney, P. Passmore, Dong-Han Ham, and P. H. Nguyen. A User Study on Curved Edges in Graph Visualization. *IEEE transactions on visualization and computer graphics*, 18(12):2449–56, dec 2012.

[21] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proceedings of the 2006 AVI workshop on BEyond time and errors novel evaluation methods for information visualization - BELIV '06*, page 1, New York, New York, USA, may 2006. ACM Press.

[22] T. Munzner. *Interactive visualization of large graphs and networks*. Doctoral dissertation, Stanford University, 2000.

[23] A. Perer and B. Shneiderman. Balancing systematic and flexible exploration of social networks. *Visualization and Computer ...*, 2006.

[24] C. Plaisant, J. Grosjean, and B. B. Bederson. SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation. pages 57–64. IEEE, 1998.

[25] E. M. Rodrigues, N. Milic-Frayling, M. Smith, B. Shneiderman, and D. Hansen. Group-in-a-Box Layout for Multi-faceted Analysis of Communities. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*, pages 354–361. IEEE, oct 2011.

[26] P. Shannon, A. Markiel, and O. Ozier. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome ...*, 2003.

[27] B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. volume 0, page 336, Los Alamitos, CA, USA, 1996. IEEE Computer Society.

[28] C. Tominski, J. Abello, and H. Schumann. CGV - An interactive graph visualization system. *Computers & Graphics*, 2009.

[29] F. Van Ham and A. Perer. "Search, show context, expand on demand": Supporting large graph exploration with degree-of-interest. In *IEEE Transactions on Visualization and Computer Graphics*, volume 15, pages 953–960, 2009.

[30] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J. D. Fekete, and D. W. Fellner. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges, 2011.